DASHBOARD
RECOMMANDATION DE RESTAURANT

Antoine PINTO

Résumé :

Sous R Shiny, création d'un dashboard de recommandation de restaurant dans lequel le client fixe des critères de filtrage de restaurant (ville, service, wifi…) auxquels il affecte des poids représentant l'importance qu'il donne aux critères. L'algorithme développé prend ainsi en compte les critères et les poids associés afin d'ajuster, au mieux, les recommandations aux préférences du client.

1. Indication d'avant lecture

    a. Accéder au Dashboard

Un Dashboard a été développé à la suite de cette étude. Vous pouvez y accéder à partir du lien suivant : https://antoine-pinto.shinyapps.io/yelp_application/. Choisissez vos propres critères de sélection et la plateforme affiche automatiquement les restaurants qui vous correspondent.

    b. Structure du dossier

Le dossier est structuré de la façon suivante :

- Le fichier *dashboard_recommandation_restaurant.pdf* contient le rapport du projet.
- Le fichier *Preprocessing.Rmd* est un fichier Rmarkdown dans lequel l'ensemble du code est rassemblé : préparation des données et implémentation du dashboard.
- Le fichier *Preprocessing.html* n'est qu'une présentation du code sous format html.
- Le dossier *code_dashboard* contient tous les fichiers nécessaires pour générer le dashboard sous R shiny.
- Le dossier *data* contient les jeux de données utilisés. Certains fichiers ont été supprimés pour cause de taille trop importante.

    c. Motivation & compétences acquises

J'ai choisi de travailler sur ce projet pour acquérir des compétences en développement de Dashboard dont je n'ai jamais eu l'occasion de programmer malgré en avoir souvent entendu l'utilité ces dernières années. Ce projet m'a également permis d'acquérir davantage de savoir-faire dans la gestion des bases de données.

2. Exploration

La réunion de l'offre et de la demande est un enjeu important en sciences économiques. Lorsque le consommateur recherche une offre qui correspond à son désir, il fait parfois face à une asymétrie d'information qui ne lui permet pas de faire le choix qui maximiserait sa satisfaction. L'asymétrie d'information prive une partie de la clientèle et impacte donc négativement le profit des entreprises.

D'autre part, même lorsque le consommateur a l'occasion de différencier les offres en fonction de ses critères personnels, il se retrouve parfois dans une situation où ses critères sont trop restrictifs et aucune offre n'y répond. Ainsi, lorsque l'on recherche son restaurant optimal, on est parfois amené à essayer plusieurs combinaisons de critère dans le but de trouver le restaurant qui correspondant au mieux à nos critères.

Cette étude développe ainsi un moyen d'ajuster rapidement les recommandations aux préférences de la clientèle par le moyen de poids d'importance associés à chaque critère.

3. Familiarisation avec le métier

Yelp est une entreprise américaine mettant en relation les consommateurs avec les commerces locaux. Les recommandations aux clients s'appuient sur les avis caractérisés comme étant « fiables » et « utiles ». La base de données Yelp est constituée de 5 tables qui hébergent chacune des informations spécifiques : *business, checkin, review, tip, user*. Dans cette étude, je n'utiliserai que les tables *business* et *review*.

La table *business* recense l'ensemble des commerces ainsi que leurs caractéristiques. Cette table est donc identifiée par la variable $business\_id$ qui représente l'identifiant de l'entreprise. 13 autres variables sont présentes :

- $business\_id$: identifiant de l'entreprise
- $adress$: adresse de l'entreprise
- $attributes$: attributs de l'entreprise (présence d'un parking, interdiction aux mineurs…)
- $categories$: catégorie de l'entreprise (restaurant…)
- $city$: ville de l'entreprise
- $hours$: horaires d'ouverture de l'entreprise
- is_open : indique si l'entreprise est encore ouverte
- $lattitude$; $longitude$
- $name$: nom de l'entreprise
- $postal\_code$: code postal de l'entreprise
- $review\_count$: nombre d'avis sur l'entreprise
- $stars$: nombre d'étoiles obtenu par l'entreprise
- $state$: état de l'entreprise

La table *review* est composée de l'ensemble des commentaires sur un an. Chaque observation s'identifie donc par la variable $review\_id$ qui représente l'identifiant du commentaire. 8 autres variables sont présentes :

- $business\_id$: identifiant de l'entreprise
- $cool$: indique si le commentaire est « cool »
- $date$: date du commentaire
- $funny$: indique si le commentaire est « funny »
- $review\_id$: identifiant du commentaire
- $stars$: nombre d'étoile attribué par l'auteur du commentaire
- $text$: commentaire
- $useful$: indique si le commentaire est utile
- $user\_id$: identifiant de l'auteur du commentaire

4. Étude de cas

a. Définition de la problématique et de la solution

L'étude de cas présenté vise à répondre à la problématique suivante : lorsqu'un client utilise une plateforme lui permettant de rechercher un restaurant, celui-ci fait un filtrage des restaurants en se basant sur des critères sélectifs. Cependant, si le client a trop de critères, alors il est possible qu'aucune offre de restaurant ne réponde à la totalité de ses clients. Le client se retrouve alors sans aucune recommandation.

Une idée naïve de réponse à ce problème d'indisponibilité de l'offre serait de proposer au client les restaurants qui valident le plus grand nombre de ses critères. Par exemple, prenons un client A qui souhaite un restaurant à Memphis, ouvert le lundi, possédant le Wifi, avec une ambiance romantique et un niveau sonore calme. Supposons qu'aucun restaurant ne réponde à la totalité de ses 5 critères. Alors l'idée naïve serait de lui recommander les restaurants qui répondent à 4 de ses critères. Ainsi, l'idée naïve revient à classer les restaurants en fonction de leur score :

$$Score_i^{NAÏF} = \sum_{j=1}^{p} \mathbb{I}(variable_j = critere_j)$$

Avec i représentant l'identifiant du restaurant, j représentant la variable et $\mathbb{I}$ l'indicatrice qui vaut 1 si la condition est vraie et 0 sinon.

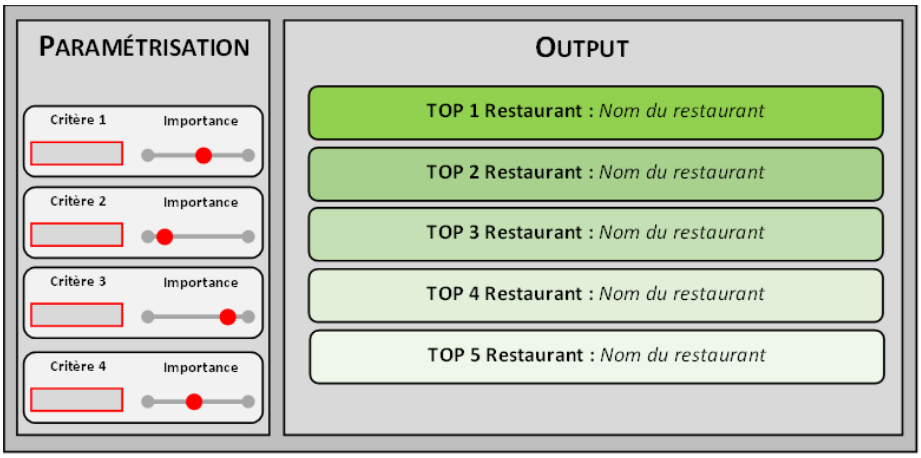
La solution présentée dans cette étude de cas est la suivante : implémenter un outil de recommandation de restaurant capable de proposer l'offre qui répond **au mieux** aux critères du consommateur. Le consommateur fixe alors une mesure d'importance à chacun de ses critères et l'algorithme s'occupe de classer les restaurants en fonction des critères et des poids associés. Puisque l'algorithme prend en compte une information supplémentaire représentant l'importance affectée à chaque critère, cette solution permet un meilleur ajustement aux préférences de la clientèle. En reprenant l'exemple du client A, il se peut que le client donne beaucoup d'importance au fait que son restaurant se trouve à Memphis et beaucoup moins au fait que le restaurant possède le Wifi. Ainsi, cette solution vise à prendre en compte l'importance que le client donne à ses critères en lui demandant, dans la paramétrisation de sa recherche, de sélectionner les poids associés à chaque critère. Cette solution vise donc à classer les restaurants en fonction du score suivant :

$$Score_i = \sum_{j=1}^{p} \mathbb{I}(variable_j = critere_j) \times poids_j$$

b. Méthodologie de la solution

La solution s'implémentera sous la forme d'un Dashboard généré à partir du package Shiny de R. Le Dashboard contiendra une fenêtre de paramétrisation et une fenêtre output. La fenêtre de paramétrisation sera constituée de tous les widgets permettant au client de choisir ses critères de filtrage ainsi que les poids associés. La fenêtre output permettra la visualisation de la sortie de l'algorithme, c'est-à-dire de l'affichage des 5 restaurants qui s'ajustent au mieux aux préférences du client.

Figure 1 : Aspect du Dashboard



Pour illustrer la méthodologie de l'algorithme basé sur les poids d'importance des critères, prenons l'exemple d'un client qui souhaite dîner dans un restaurant et paramétrise sa recherche de la façon suivante :

Tableau 1: Paramétrisation du client

Variable	Critère	Poids
Ville	New-York	80
Service	Dîner	60
Wifi	Oui	50
Ambiance	Fun	20

Supposons que nous n'ayons que 4 restaurants dans notre base de données. Aucun de ces 4 restaurants ne remplit l'intégralité des 4 critères du client. Cependant, comme nous avons une information sur l'importance que le client attribue à chaque critère, nous calculons, à partir des étapes présentées dans le tableau suivant, que le restaurant qui s'ajuste le mieux aux désirs du client est le restaurant numéro 3.

Tableau 2: Calcul du score

	BASE DE DONNÉES INITIALE				VALIDATION DES CRITÈRES				SCORE PAR VARIABLE				SCORE TOTAL
Identifiant restaurant	Ville	Service	Wifi	Ambiance	Ville	Service	Wifi	Ambiance	Ville	Service	Wifi	Ambiance	Score
1	New York	Déjeuner	Oui	Fun	Ok	No	Ok	Ok	80	0	50	20	150
2	Memphis	Déjeuner	Oui	Normale	No	No	Ok	No	0	0	50	0	50
3	New York	Dîner	Oui	Calme	Ok	Ok	Ok	No	80	60	50	0	190
4	Austin	Dîner	Non	Calme	No	Ok	No	No	0	60	0	0	60

5. Implémentation de la solution

Afin de gérer la base de données et de créer le Dashboard, j'utilise le logiciel R et notamment le package Shiny. Une partie importante de l'étude est allouée à la préparation des données puisque certaines variables doivent être recodée, notamment les variables $attributs$ et $hours$ qui sont constituées de listes et de sous-listes comme illustré sur la figure suivante :

Figure 2: Extrait de la table business

business_id	attributes	hours
6iYb2HFDywm3zjuRg0shjw	Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol...	Row(Friday='11:0-23:0', Monday='11:0-23:0', Saturday='11:0...
tCbdrRPZA0oiIYSmHG3J0w	Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol...	Row(Friday='5:0-18:0', Monday='5:0-18:0', Saturday='5:0-18...
bvN78flM8NLprQ1a1y5dRg	Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol...	Row(Friday='11:0-18:0', Monday=None, Saturday='11:0-18:...
oaepsyvc0J17qwi8cfrOWg	Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol...	
PE9uqAjdw0E4-8mjGl3wVA	Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol...	Row(Friday='16:0-19:0', Monday='16:0-19:0', Saturday='9:0-...
D4JtQNTI4X3KcbzacDJsMw	Row(AcceptsInsurance=None, AgesAllowed=None, Alcohol...	Row(Friday='17:0-21:0', Monday='17:0-21:0', Saturday='17:0...

Afin d'implémenter un Dashboard agréable à manipuler, la préparation de la base de données a également eu pour objectif de préparer seulement les variables intéressantes pour un client. Ainsi, la fenêtre « paramétrisation » du Dashboard permet au client de jouer avec les critères suivants :

- Mots-clés : choix d'un mot clé qui définit le restaurant.
- Jour : choix du jour d'ouverture du restaurant.

- Heure : choix de l'horaire d'ouverture du restaurant.
- Ville : choix de la ville du restaurant.
- Importance des avis : degré d'importance que le client donne à la popularité du restaurant.
- Niveau des prix : choix d'un intervalle entre 0 et 3 représentant le niveau de prix que le client désire.
- Service : choix du service (lunch/dinner ou breakfast/brunch)
- Ambiance : choix de l'ambiance du restaurant (casual, hipster, romantic, upscale)
- Niveau du bruit. Choix d'un intervalle entre 1 et 4 représentant le niveau de bruit du restaurant.
- Autres attributs : choix d'autres paramètre (Bon pour danser ; Carte de crédit acceptée ; Chiens autorisés ; accessible en fauteuil roulant…)

Parmi les critères, seuls 3 sont restrictifs, c'est-à-dire qu'il filtre la base de données sans considérer le critère d'importance : mots-clés ; jour et heure. Si le mot-clé n'est présent dans la description d'aucun restaurant, alors l'algorithme ne prend pas en compte le mot-clé.

Le critère d'importance des avis représente le degré auquel le client donne de l'importance à la popularité de chaque restaurant. En effet, la popularité d'un restaurant est calculée est fonction du nombre d'étoiles de ses avis ainsi que le nombre d'avis publiés de la façon suivante :

$$popularité_i = étoile_i \times \log(nombre\ d'avis)$$

Le score de chaque restaurant est recalculé à chaque fois que le client effectue une modification dans son paramétrage. Ainsi, l'algorithme va classer les restaurants et afficher dans la fenêtre « output » les 5 restaurants qui correspondent le plus aux préférences du client. Étant donné que la variable $popularité$ est un nombre réel, il est très peu probable que deux restaurants obtiennent le même score. Cela est davantage probable si le client paramétrise l'importance des avis à 0.

a. Indicateur de pureté

Un indicateur de pureté d'ajustement aux préférences du consommateur peut également être calculé. En effet, le score maximum qu'un restaurant peut avoir, s'il remplie tous les critères demandés par le client, est donné par la formule suivante :

$$ScoreMax = \sum_{j=1}^{p} poids_j$$

Ainsi l'indicateur de pureté d'ajustement aux préférences pour un restaurant donné peut se calculer comme son score obtenu divisé par le score maximum possible :

$$Pureté = \frac{Score_i}{ScoreMax}$$

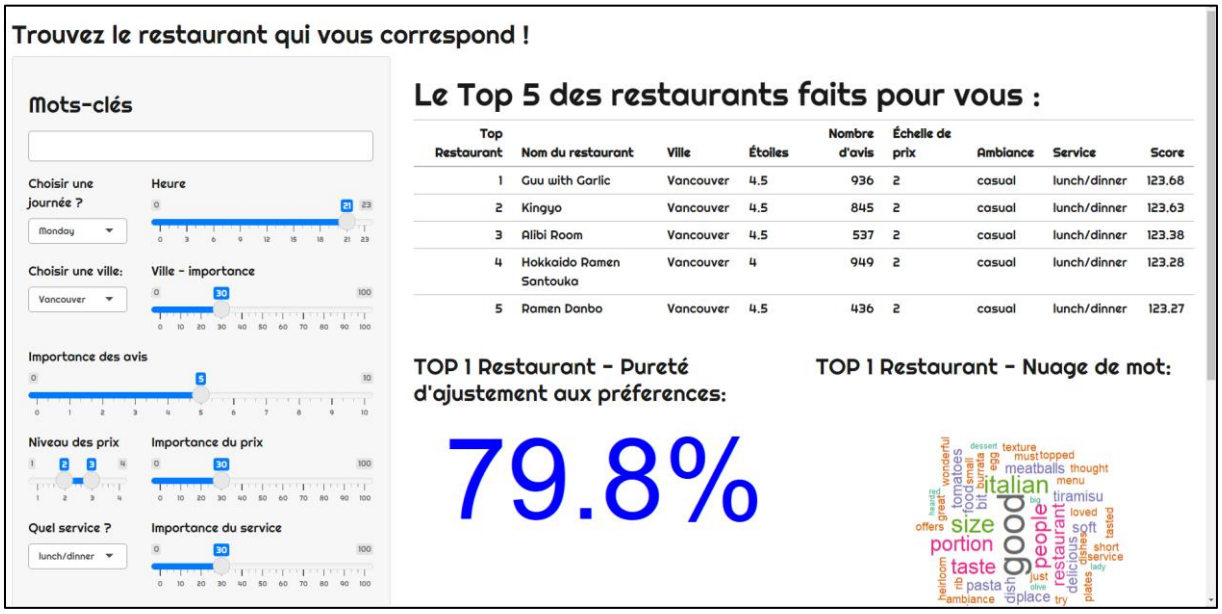
b. Nuage de mot

Pour permettre au client d'avoir une meilleure appréhension du restaurant que l'algorithme lui propose, un nuage de mot est affiché. Il permet de mettre en avant les mots-clés les plus fréquents dans l'ensemble des avis du restaurant de la table $review$. Ainsi le nuage de mot du meilleur restaurant, celui correspondant au mieux aux préférences du client, est affiché dans la fenêtre « output ».

c. Création du Dashboard

Le fichier contenant le Dashboard est disponible en pièce jointe de cette présentation. Ci-dessous une capture d'écran du Dashboard :

Figure 3: Dashboard sous R Shiny



6. Analyse

La solution présentée dans cette étude possède deux avantages majeurs. Le premier avantage est qu'elle permet d'éviter le problème d'indisponibilité de l'offre due au fait que le client possède des critères trop restrictifs qu'aucun restaurant ne respecte entièrement. Le système de recommandation basé sur un classement permet toujours de proposer quelque chose au client, quelque soit ses critères (non restrictifs). Le second avantage est que cette solution permet d'ajuster, au mieux, les recommandations aux préférences des clients grâce à l'utilisation de la mesure d'importance des critères.

Cependant, le désavantage majeur de cette solution est qu'elle nécessite 2 fois plus de paramètres à fixer pour le client, ce qui peut provoquer de l'impatience par exemple.

7. Conclusion - RoadMap pour la suite

En plus des avantages cités précédemment, cette solution permet également de capter des informations très importantes sur les préférences des consommateurs. En effet, les données concernant les critères souhaités par un client ainsi que l'importance qu'il donne à ces critères pourront être utilisés ultérieurement par Yelp afin de d'effectuer de la recommandation au client, même en dehors d'une recherche de restaurant.

Yelp va également pouvoir étudier la corrélation entre les préférences à certains critères. En effet, il se peut que, par exemple, les clients qui sélectionne le critère « GoodForKids » sont aussi ceux qui sélectionnent souvent le critère « Wifi ». Ainsi Yelp pourrait recommander aux restaurant accueillant des enfants d'investir dans une borne Wifi pour sa clientèle.

Grâce au critère relatif au prix, Yelp pourra également faire de la différentiation de prix. En effet, ce critère permettra de connaître le profil des personnes qui sont prêtes à payer le prix fort en fonction de leurs caractéristiques liées aux autres critères et ainsi pour proposer aux restaurants de faire une

différentiation par les prix en fonction du type de consommateur comme un prix étudiant ou alors de faire payer certains services comme le Wifi.

Cette méthode va également permettre à Yelp de savoir ce qu'il manque à un restaurant pour qu'il corresponde mieux aux critères demandés par le client. Par exemple, si Yelp s'aperçoit que pour un nombre important de clients, un restaurant est proposé mais n'atteint pas les 100% d'indice de pureté parce qu'un critère demandé par le client, le Wifi, n'est pas présent dans le magasin mais intéresse beaucoup de clients qui en ont fait la recherche. Grâce à cette information, Yelp va pouvoir recommander au magasin d'installer le Wifi car c'est un désir de la plupart de ses clients.

Figure 4: RoadMap